# Отчет о проведении тестирования.

Время проведения: *15.05.2021*

## 1. Цель теста

Бизнес-цель теста – *получение навыков работы с репликацией БД и нагрузочного тестирования.*

Техническая цель теста – оценка производительности с использованием *wrk* и *СУБД MySQL 8.0.24* на кластере *Kubernetes v1.21.0*.

*В испытании участвовало 2 операций.*

*Таблица 1. Модель нагрузки.*

| | |
|---|---|
| *1* | *Нагрузка на систему с одним инстансом БД MySQL.* |
| *2* | *Нагрузка на систему с тремя инстансами (master + 2 slave) БД MySQL.* |

*Было произведено 8 экспериментов с разным количеством запросов в секунду.*

*Таблица 2. Эксперименты.*

| | |
|---|---|
| *1* | *1 поток 1 соединение* |
| *2* | *2 потока 10 соединений* |
| *3* | *4 потока 100 соединений* |
| *4* | *8 потоков 1000 соединений* |

## 2. Выводы

1. Максимальная производительность системы – *37.41* запроса в секунду при 4-х потоковом тесте и *100* соединениях.
2. Время *отклика / выполнения* на уровне максимальной производительности по операциям составляло от *3.25* секунд до *7,48* секунды. Так же были просадки после *90 перцентиля* в диапазоне *7-14* секунд.
3. Количество операций в секунду выросло пропорционально количеству нод.

### Настройки, создание кластера master/slave

1. Для настройки репликации *master/slave* использовались возможности кластера *Kubernetes*.
2. Развертывание и клонирование БД осуществлялось скриптом *configure-slave.sh*.

```
if [[ "x$1" == "xrun" ]] ; then
  (/bin/bash $0)&
  exit 0
fi
# Skip the clone if data already exists.
[[ -f /var/lib/mysql/mysql/done ]] && exit 0
# Skip the clone on primary (ordinal index 0).
[[ `hostname` =~ -([0-9]+)$ ]] || exit 0
ordinal=${BASH_REMATCH[1]}
[[ $ordinal -eq 0 ]] && exit 0
echo "Waiting for mysqld to be ready (accepting connections)"
c=1; until mysql -h 127.0.0.1 -e "SELECT 1" || [[ $c -le 6 ]] ; do sleep $c; done
sleep $c
# Clone data from previous peer.
echo "Clone data from previous peer."
mysql -h 127.0.0.1 -e "STOP SLAVE"
mysql -h 127.0.0.1 -e "RESET SLAVE"
mysql -h 127.0.0.1 -e "RESET SLAVE ALL"
mysqldump -h mysql-0.mysql.default.svc.cluster.local \
          --all-databases --master-data --triggers --routines --events |
    mysql -h 127.0.0.1
mysql -h mysql-0.mysql.default.svc.cluster.local -e 'SHOW MASTER STATUS' |
               awk               '/mysql-0/{printf"CHANGE               MASTER               TO
MASTER_HOST=\"mysql-0.mysql.default.svc.cluster.local\"'\
                    ',MASTER_LOG_FILE=\"%s\"'\
                    ',MASTER_LOG_POS=%s'\
                    ',MASTER_USER=\"root\"'\
                    ',MASTER_PASSWORD=\"\"'\
                    ',MASTER_CONNECT_RETRY=10\n",$1,$2}' |
 mysql -h 127.0.0.1
echo "Clone done."
```

3. Настройка Single node производилась при при значении *replicas=1*, клонирование БД в этой конфигурации не производилась так как предусмотрено в скрипте:

```
# Skip the clone if data already exists.
[[ -d /var/lib/mysql/mysql ]] && exit 0
# Skip the clone on master (ordinal index 0).
[[ `hostname` =~ -([0-9]+)$ ]] || exit 1
```

4. Файл описания *StatefulSet-a*: *mysql-statefulset.yaml*,
   файл описания сервисов: *mysql-services.yaml*,
   файл описания дополнительной конфигурации: *mysql-configmap.yaml*.
5. Балансировка нагрузки на чтение, также осуществлялась при помощи кластера *Kubernetes*. С использованием сервиса *Kubernetes*. Файл описания *mysql-reads.loadBalancer.yaml*.
6. При проведении нагрузочного тестирования наиболее ресурсоемкие запросы:

```
SELECT u.id, username, name, surname, age, sex, city,
       JSON_ARRAYAGG(interests), NOT isnull(uhf.id) AS is_friend
 FROM user u
 LEFT JOIN user_has_interests uhi ON uhi.user_id = u.id
 LEFT JOIN interest i ON i.id = uhi.interest_id
 LEFT JOIN user_has_friends uhf ON uhf.friend_id = u.id
  AND uhf.user_id = 0x...
 WHERE name LIKE '?%%' AND surname LIKE '?%%'
 GROUP BY u.id, username, name, surname, age, sex, city, uhf.id;
```

и

```
SELECT u.id, username, name, surname, age, sex, city,
       JSON_ARRAYAGG(interests), NOT isnull(uhf.id) AS is_friend
 FROM user u
 LEFT JOIN user_has_interests uhi ON uhi.user_id = u.id
 LEFT JOIN interest i ON i.id = uhi.interest_id
 LEFT JOIN user_has_friends uhf ON uhf.friend_id = u.id
  AND uhf.user_id = 0x...
 WHERE surname LIKE '?%%'
 GROUP BY u.id, username, name, surname, age, sex, city, uhf.id;
```

# 3. Графики

## 3.1. График latency и Throughput.

*График 1. latency (ms).*

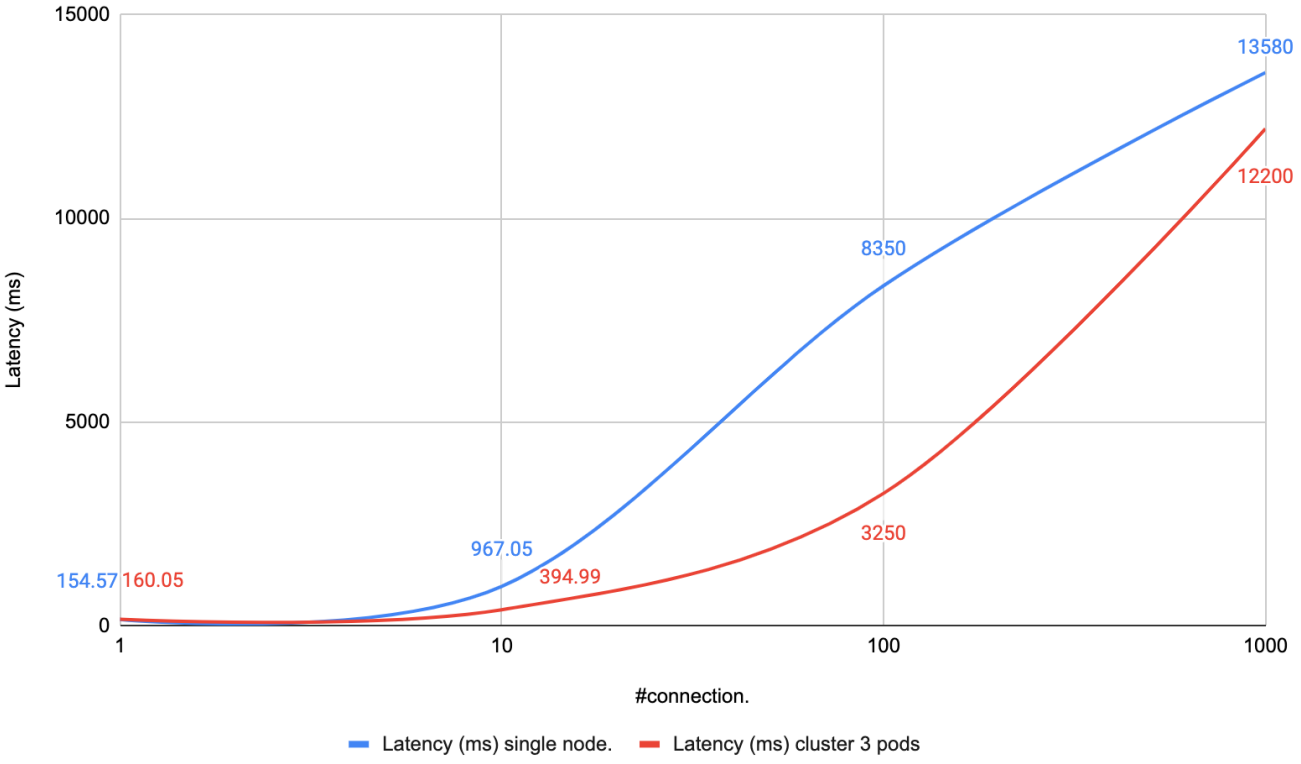
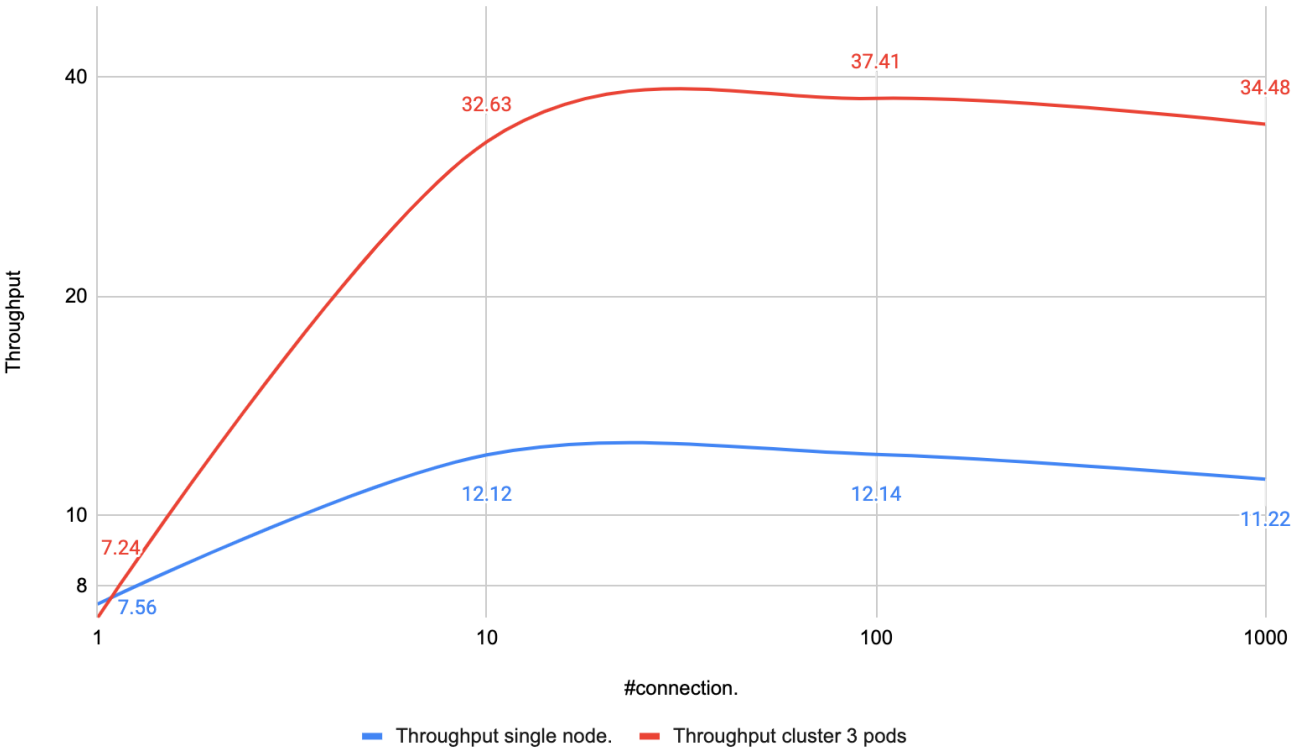
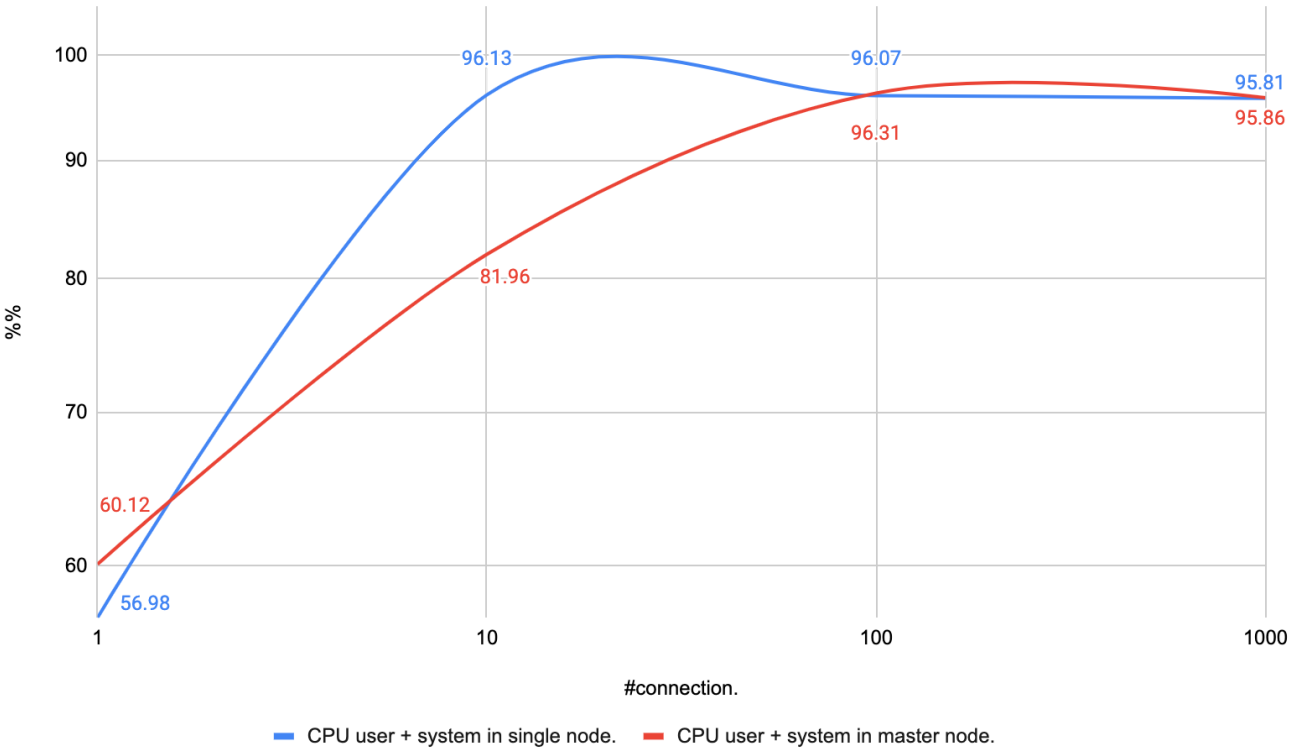


*График 2. Throughput.*




## 3.2. Нагрузка CPU.

*График 3. Нагрузка CPU.*



## 3.3. Load averages.

*График 5. График LA 1m.*

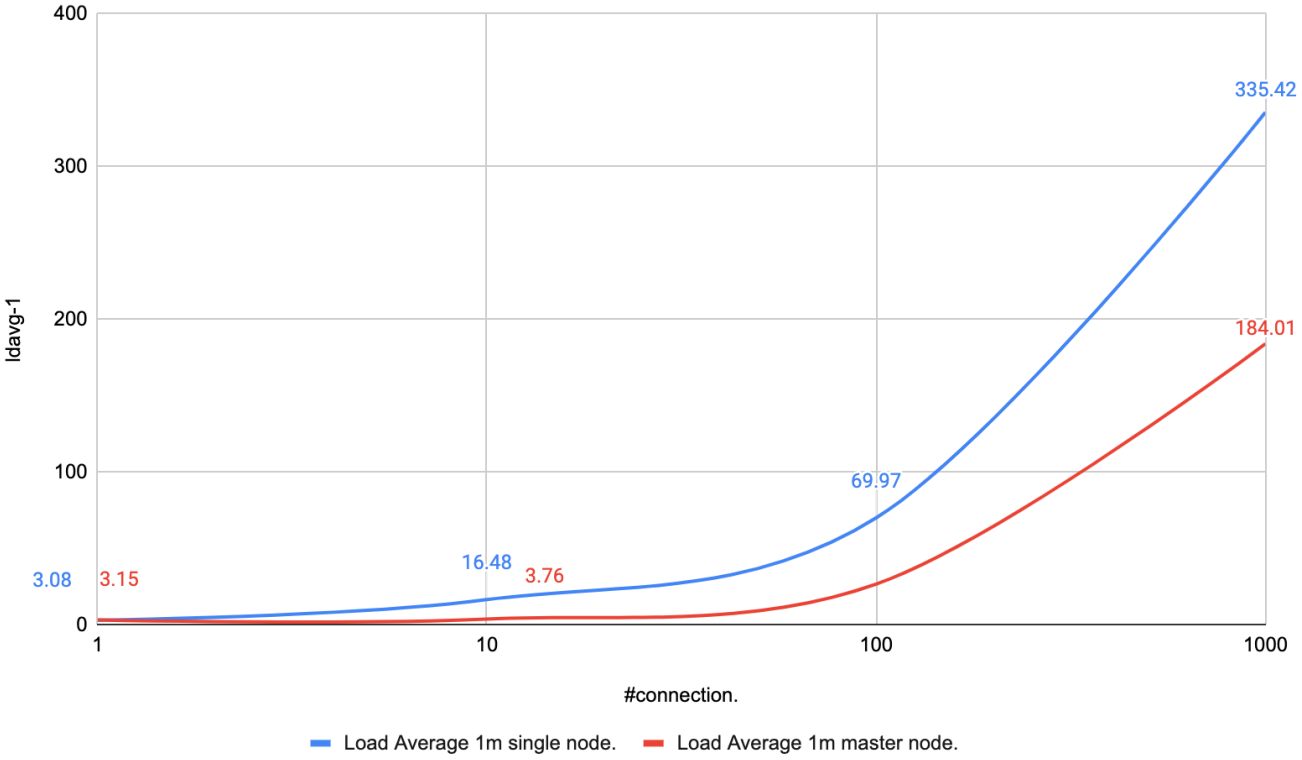
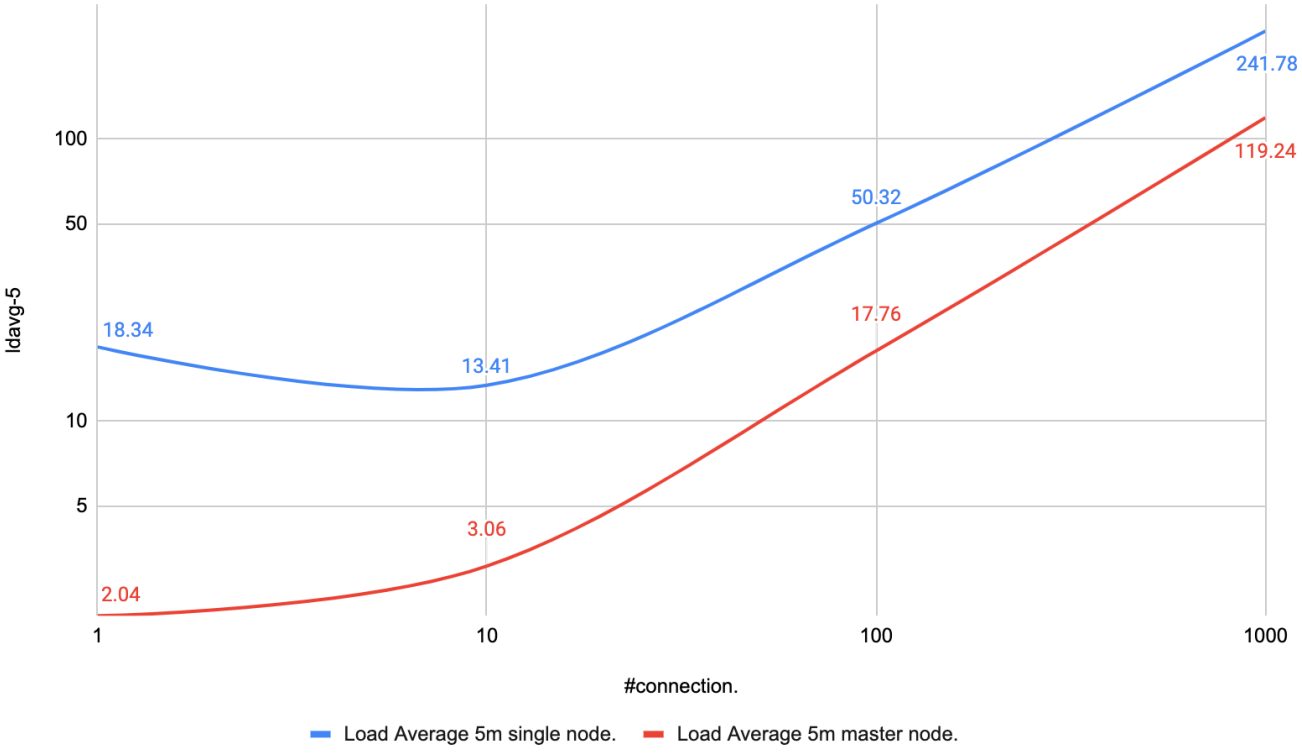



*График 6. График LA 5m*



## 3.4. I/O - disc usage.

*График 8. Количество дисковых операции в секунду и поток чтения. Single node.*

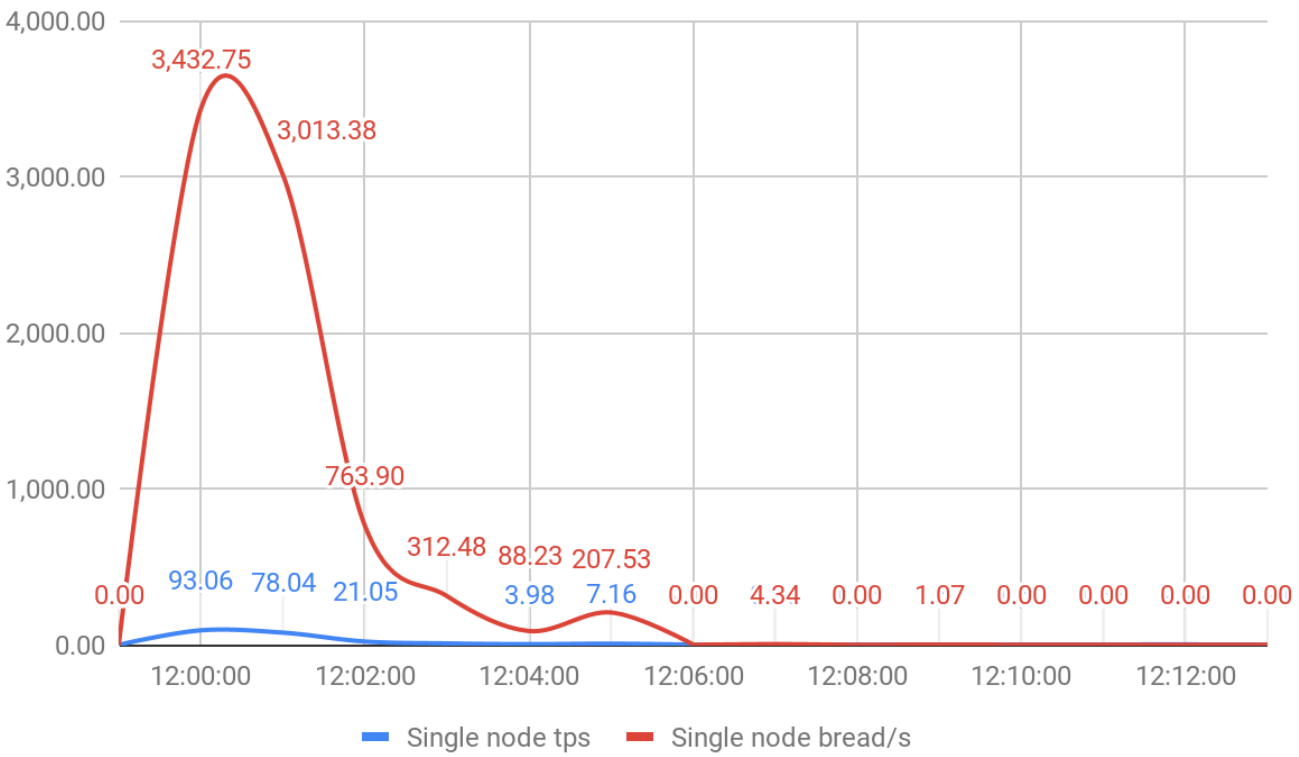


*График 9. Количество дисковых операции в секунду и поток чтения. Cluster 3 pods master node.*